

Practical: 8

AIM- What is BroadcastReceiver? Difference between Service and BroadcastReceiver. Create an Alarm application by using service & BroadcastReceiver by following below instructions.

Submitted By :- Ghadiya Harshil
Enrollment number :- 21012021031



**Ganpat
University**

॥ विद्यया समाजोत्कर्षः ॥

**U.V. Patel
College of
Engineering**

Practical: 8

AIM: What is BroadcastReceiver? Difference between Service and BroadcastReceiver.

Create an Alarm application by using service & BroadcastReceiver by following below instructions.

1. Create MainActivity according to below UI design.
2. Create AlarmBroadcastReceiver class
3. Create AlarmService Class
4. Add android.permission.SCHEDULE_EXACT_ALARM Permission in ManifestFile

- What is BroadcastReceiver ?

Broadcast in android is the system-wide events that can occur when the device starts, when a message is received on the device or when incoming calls are received, or when a device goes to airplane mode, etc.

Difference between Service and BroadcastReceiver.

Service : If you want to do something in background , this will be running always in background even if the application closed. You can create this in separate process and also you can give your service to other app if you want. Downloading any content or Music is good example.

Broadcast Reciever : Usually system will send some info which can be recieved by your app if you would wish to ,by registering. And you can do something what you want when that thing happens by using onReceive method. Example is the system will send BroadcastReceiver when new sms arrives or Booting done

Activity main.xml :-

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<androidx.core.widget.NestedScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">
```

```
<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:layout_marginStart="60dp"
        android:layout_marginTop="44dp"
        android:text="@string/app_name"
        android:textSize="25sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <com.google.android.material.card.MaterialCardView
        android:id="@+id/card"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        app:cardElevation="20dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <ImageView
                android:id="@+id/imageView"
                android:layout_width="match_parent"
                android:layout_height="194dp"
                android:scaleType="fitXY"
                app:srcCompat="@drawable/clock" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"
                android:padding="16dp">

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
```

Practical: 8

```
        android:text="Create Alarm Time"
        android:textAppearance="?attr/textAppearanceHeadline6" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="Be pressing buttons,Alarm can be created And cancelled."
    android:textAppearance="?attr/textAppearanceBody2" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="Check Current Time By looking below real digital clock"
    android:textAppearance="?attr/textAppearanceBody2" />

<TextClock
    android:id="@+id/clocktime"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="20dp"
    android:format12Hour="hh:mm:ss a MMM dd yyyy"
    android:padding="10dp"
    android:textAlignment="center" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/create"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Create Alarm"
    app:icon="@drawable/baseline_add_alarm_24"/>
</LinearLayout>
</LinearLayout>

</com.google.android.material.card.MaterialCardView>

<com.google.android.material.card.MaterialCardView
    android:id="@+id/card2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    app:cardElevation="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/card">
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="match_parent"
        android:layout_height="194dp"
        android:scaleType="fitXY"
        app:srcCompat="@drawable/clock"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Set Alarm Time:"
            android:layout_gravity="center"
            android:textAppearance="?attr/textAppearanceHeadline6"/>

        <TextClock
            android:id="@+id/clocktime2"
            android:format12Hour="hh:mm:ss a MMM dd yyyy"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_margin="20dp"
            android:padding="10dp"
            android:textAlignment="center"/>

        <com.google.android.material.button.MaterialButton
            android:id="@+id/cancel"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Cancel Alarm"
            app:icon="@drawable/baseline_alarm_off_24"/>
    </LinearLayout>
</LinearLayout>

</com.google.android.material.card.MaterialCardView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Practical: 8

</androidx.core.widget.NestedScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

MainActivity.kt :-

```
package com.example.mad_practical_8_21012021031

import android.app.AlarmManager
import android.app.PendingIntent
import android.app.TimePickerDialog
import android.content.Intent
import android.os.Bundle
import android.view.View
import android.widget.TextClock
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.example.mad_practical_8_21012021031.AlarmBroadcastReceiver
import com.google.android.material.button.MaterialButton
import com.google.android.material.card.MaterialCardView
import java.text.SimpleDateFormat
import java.util.Calendar

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val addAlarm : MaterialButton = findViewById(R.id.create)
        val card : MaterialCardView = findViewById(R.id.card2)

        card.visibility = View.GONE

        addAlarm.setOnClickListener {
            TimePickerDialog(this, { tp, hour, minute -> setAlarmTime(hour, minute) },
            Calendar.getInstance().get(Calendar.HOUR), Calendar.getInstance().get(Calendar.MINUTE),
            false).show()
            card.visibility = View.VISIBLE
        }

        val cancelAlarm : MaterialButton = findViewById(R.id.cancel)
        cancelAlarm.setOnClickListener {
            stop()
            card.visibility = View.GONE
        }
    }
}
```

```
}  
}  
  
fun setAlarmTime(hour : Int, minute : Int){  
    val alarmtime = Calendar.getInstance()  
    val year = alarmtime.get(Calendar.YEAR)  
    val month = alarmtime.get(Calendar.MONTH)  
    val date = alarmtime.get(Calendar.DATE)  
    alarmtime.set(year, month, date, hour, minute, 0)  
    val textAlarmTime : TextClock = findViewById(R.id.clocktime2)  
    textAlarmTime.text = SimpleDateFormat("hh:mm:ss a").format(alarmtime.time)  
    setAlarm(alarmtime.timeInMillis, AlarmBroadcastReceiver.ALARMSTART)  
    Toast.makeText(this, "Alarm Set", Toast.LENGTH_LONG).show()  
}  
  
fun stop() {  
    setAlarm(-1, AlarmBroadcastReceiver.ALARMSTOP)  
}  
  
fun setAlarm(militime : Long, action : String) {  
    val intentalarm = Intent(applicationContext, AlarmBroadcastReceiver::class.java)  
    intentalarm.putExtra(AlarmBroadcastReceiver.ALARMKEY, action)  
    val pendingintent = PendingIntent.getBroadcast(applicationContext, 4356, intentalarm,  
        PendingIntent.FLAG_UPDATE_CURRENT or PendingIntent.FLAG_IMMUTABLE)  
    val alarmmanager = getSystemService(ALARM_SERVICE) as AlarmManager  
  
    if (action == AlarmBroadcastReceiver.ALARMSTART) {  
        alarmmanager.setExact(AlarmManager.RTC_WAKEUP, militime, pendingintent)  
    }  
    else if (action == AlarmBroadcastReceiver.ALARMSTOP) {  
        alarmmanager.cancel(pendingintent)  
        sendBroadcast(intentalarm)  
    }  
}  
}
```

AlarmBroadcastReceiver.kt

```
package com.example.mad_practical_8_21012021031
```

```
import android.content.BroadcastReceiver  
import android.content.Context  
import android.content.Intent
```

```
class AlarmBroadcastReceiver : BroadcastReceiver() {  
    companion object {  
        val ALARMKEY = "KEY"
```

Practical: 8

```
val ALARMSTART = "START"
val ALARMSTOP = "STOP"
}

override fun onReceive(context: Context, intent: Intent) {
    if (intent != null) {
        val data = intent.getStringExtra(ALARMKEY)

        if (data != null) {

            val intentservice = Intent(context, AlarmService::class.java)

            if (data == ALARMSTART) {
                context.startService(intentservice)
            }
            else{
                context.stopService(intentservice)
            }
        }
    }
}
```

AlarmService.kt

```
package com.example.mad_practical_8_21012021031

import android.app.Service
import android.content.Intent
import android.media.MediaPlayer
import android.os.IBinder

class AlarmService : Service() {

    lateinit var player : MediaPlayer
    override fun onBind(intent: Intent): IBinder {
        TODO("Return the communication channel to the service.")
    }

    override fun onDestroy() {
        player.stop()
        super.onDestroy()
    }

    override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {
        if (intent!=null){
```


Practical: 8

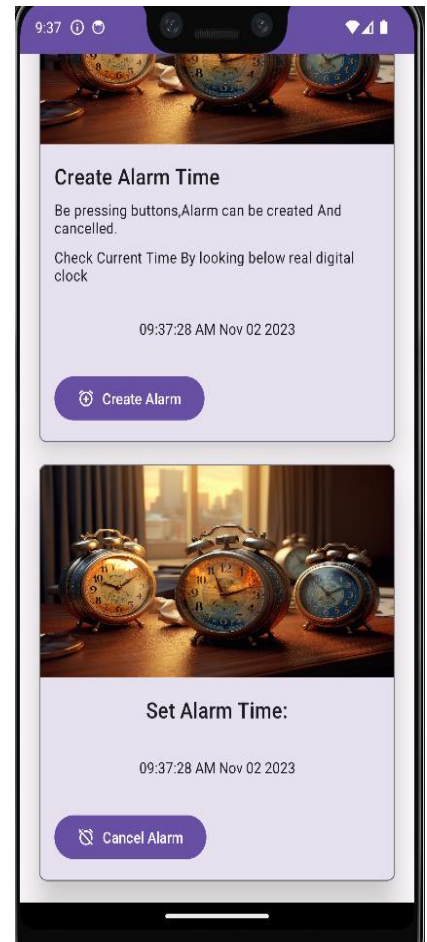
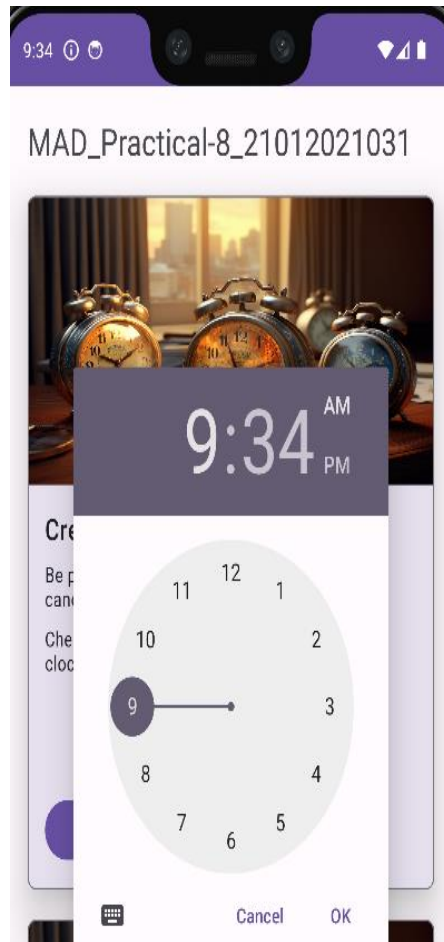
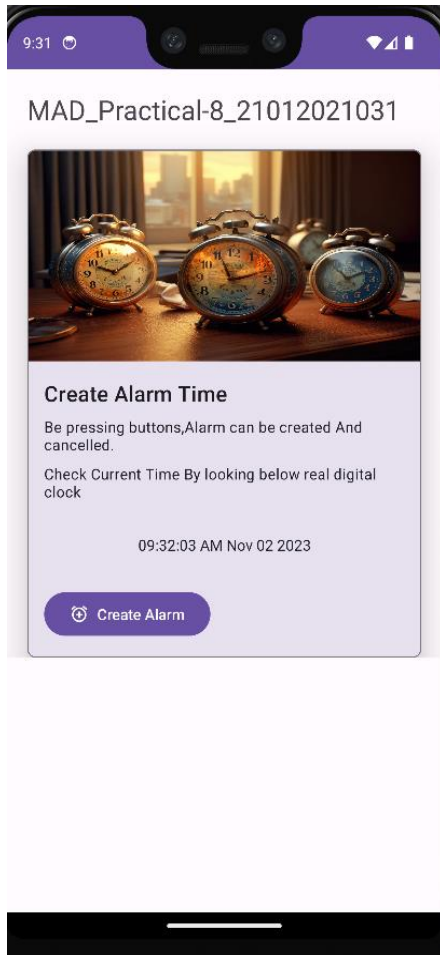
```
        player = MediaPlayer.create(this,R.raw.alarm)
    }
    return START_STICKY
}
}
```

AndroidManifest.xml

```
<uses-permission android:name="android.permission.SCHEDULE_EXACT_ALARM"/>
```

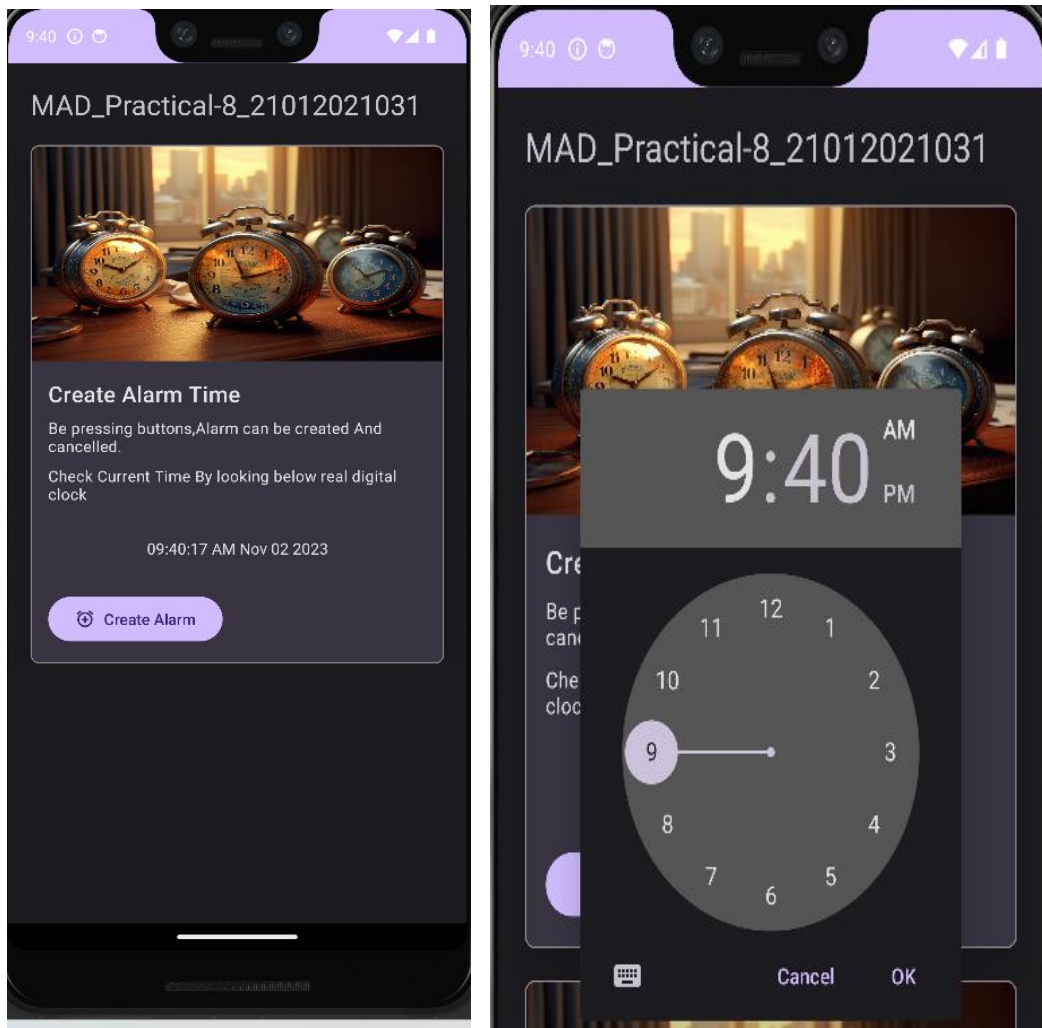
Output :-

LIGHT THEME



Practical: 8

Dark Theme:-



Practical: 8

